کاربرد یادگیری تقویتی در جهت‌یابی حرکت سرپنتین ربات مارمانند*

هادی کلانی[۱] علیرضا اکبرزاده[۲]

چکیده *هدف این مقاله هدایت و کنترل ربات مارمانند با استفاده از یادگیری تقویتی (Reinforcement learning) می‌باشد. این مقاله به مدل‌سازی و شبیه‌سازی ربات مارمانند می‌پردازد و سپس نتایج آن را روی ربات واقعی پیاده‌سازی می‌نماید. در ابتدا معادلات دینامیک یک ربات مارمانند با n عضو (Link) در حرکت سرپنتین (Serpentine Locomotion) با استفاده از روش گیبس-اپل (Gibbs-Appell) به‌صورت ساده، جامع و کارآمد به‌دست آمده است. روش ارائه شده در این مطالعه حجم محاسبات برای دینامیک ربات مارمانند را تا حد قابل توجهی نسبت به‌کارهای پیشین کاهش می‌دهد. سپس مدل فیزیکی ربات در نرم‌افزار سیم‌مکانیک (SimMechanics) مدل‌سازی گشته و از آن برای تأیید معادلات دینامیک استفاده شده است. در این مقاله از یادگیری Q (Q learning) برای آموزش ربات مارمانند و جهت‌یابی آن استفاده شده است. هم‌چنین تأثیر پارامترهای منحنی سرپنوید و منحنی بدن مار روی سرعت یادگیری بررسی شده است. نتایج نشان می‌دهند پارامترهای فیزیکی که شکل ربات مار را تغییر نمی‌دهند، بر روی یادگیری ربات تأثیر محسوسی نمی‌گذارند. در انتها، نرم‌افزار شبیه‌سازی وباتس و ربات مارمانند FUM-Snake II برای تصدیق نتایج حاصل از یادگیری به‌کار گرفته شده است. نتایج به‌دست آمده از ربات آزمایشگاهی نشان می‌دهد که مسیر حرکت این ربات پس از یادگیری به روش Q، با مسیر پیش‌بینی شده توسط حل دینامیکی پیشنهادی و نتایج شبیه‌سازی در نرم‌افزار وباتس همگی دارای مطابقت خوبی می‌باشند.*

واژه‌های کلیدی ربات مارمانند، روش گیبس-اپل، یادگیری تقویتی، یادگیری Q.

Application of Reinforcement Learning for Navigation of a Planar Snake Robot in Serpentine Locomotion

H. Kalani A. Akbarzadeh

Abstract *This article presents an implementation of a reinforcement learning (RL) method for a snake like robot navigation. The paper starts with developing kinematics and dynamics model of a snake robot in serpentine locomotion followed by performing simulation and finishes with actual experimentation. First, Gibbs-Appell's method is used to obtain the robot dynamics. The robot is also modeled in SimMechanics toolbox of MATLAB software which is then used to verify the derived dynamics equations. In this study, for the first time, Q-learning method is employed to obtain the optimal state and actions. Effects of serpenoid curve and body curve parameters on the snake robot learning ability are also investigated. Results indicate that parameters which do not affect body shape of the snake robot, also do not affect the learning ability. Finally, the experimental FUM-Snake II as well as webots software are both employed to validate theoretical results. Results show that the Q-learning method is an effective method for navigation of snake like robot.*

Key Words Snake robot, Gibbs–Appell's method, Reinforcement learning, Q-learning.

*تاریخ دریافت مقاله ۹۱/۱/۳۱ و تاریخ پذیرش آن ۹۲/۱۱/۳۰ می‌باشد.

(۱) نویسندهٔ مسئول، دانشجوی دکتری، گروه مکانیک، دانشکدهٔ مهندسی، دانشگاه فردوسی مشهد. Hadi.kalani@yahoo.com

(۲) استاد، گروه مکانیک، دانشکدهٔ مهندسی، دانشگاه فردوسی مشهد.

مقدمه

ربات‌های مارمانند اولین بار توسط هیروس [1] در اوایل سال ۱۹۷۰معرفی شده‌اند. ربات‌های مارمانند برای انجام دادن مأموریت‌هایی مانند اکتشاف، عملیات نجات و عملیات بازرسی پیشنهاد شده‌اند. اکثر ایده‌پردازی‌ها در مورد حرکت این ربات‌ها با الهام‌پذیری از طبیعت بوده‌است. حرکت سرپنتین عام‌ترین حرکت مار در طبیعت است. سایتو و همکارانش [2] رباتی مارمانند را که توانایی حرکت در محیط‌های مختلف داراست ساختند و معادلات حرکت سرپنتین را به‌دست آوردند. ترنست [3] به بررسی سینماتیک و دینامیک ربات مارمانند در دو بعد و سه بعد پرداخته است. به‌علاوه ترنست و همکارانش [4] گزارشی در مورد روش‌های سینماتیکی و دینامیکی ربات‌های مارمانند تا زمان خود ارائه دادند. وثوقی و همکارانش [5] ساختاری جدید برای ربات مارمانند ارائه نمودند و در مقالهٔ خود برای محاسبهٔ دینامیک از روش گیبس–اپل (Gibbs–Appell) استفاده کردند. در ادامهٔ تحقیق بر روی ربات مارمانند، اسپرانکلین و همکارانش [6] حرکت دوبعدی مستقیم‌الخط و لیلجبک و همکارانش [7] حرکت پهلویی را که یک حرکت سه‌بعدی است مورد تحلیل قرار دادند. هم‌چنین آن‌ها از فنر و دمپر برای ایجاد اصطکاک در خلال حرکت استفاده کرده‌اند. یه و همکارانش [8] به جزئیات طراحی ربات مارمانند که دارای سرعت بالایی می‌باشد، پرداختند. هم‌چنین آن‌ها حرکت سرپنتین را به‌کمک این ربات مورد بررسی قرار دادند. کرسپی و همکارانش [9] نشان دادند که برای این‌که ربات مارمانند در محیط‌های مختلف قادر به حرکت باشد، نیاز به الگویی دارد که بتواند با محیط اطراف خود وفق یابد. به‌عنوان مثال آن‌ها نشان دادند که برای ایجاد حرکت شناکردن (Swimming)، به یک موج‌های سینوسی با اختلاف فاز پایین نیاز است. مقالات متعددی از روش‌های تکاملی مانند الگوریتم ژنتیک برای یادگیری و تعلیم ربات استفاده کرده‌اند. حسن‌زاده و اکبرزاده [10,11] با استفاده از الگوریتم ژنتیک حرکت جدیدی را به نام (Forward Head Serpentine) FHS ایجاد کردند. این حرکت به‌کمک روش بهینه‌سازی الگوریتم ژنتیک، به‌گونه‌ای انجام شده‌است که عضو اول (به‌عنوان دم ربات) همواره دارای جهتی ثابت و به سمت جلو باشد. هدف از ایجاد این حرکت قرار دادن دوربین بر روی عضو اول ربات مارمانند است. اکبرزاده و همکارانش [12,13] حرکت موجی شکل را از نظر سینماتیکی و دینامیکی مورد تحلیل قرار دادند. آن‌ها در تحلیل خود از دو منحنی سرپنوید متقارن و نامتقارن استفاده کردند. آن‌ها نشان دادند که با استفاده از منحنی نامتقارن علی‌رغم افزایش سرعت پیش‌روی ربات، گشتاور مصرفی نسبت به حالت منحنی متقارن، تغییر چندانی نمی‌کند. آن‌ها [14] روش سینماتیکی جدیدی برای ایجاد حرکت موجی شکل ارائه دادند. هم‌چنین آن‌ها [15] با استفاده از طراحی آزمایش‌ها، DOE (Design of experiment)، پارامترهای تأثیرگذار در عملکرد ربات مارمانند را بررسی کردند و با الگوریتم SA (Simulated annealing) عملیات بهینه‌سازی را انجام دادند.

در مسأله یادگیری تقویتی (Reinforcement learning)، RL، به عامل (Agent) این امکان داده می‌شود که ضمن تعامل با محیط و کسب تجربه، با گسترش اعمالی که منجر به پیامد مطلوب می‌شود و محدود کردن اعمالی که منجر به پیامد نامطلوب می‌گردد، به استراتژی و سیاست بهینه برای رسیدن به هدف دست یابد [16]. برخی از ویژگی‌های روش‌های یادگیری تقویتی عبارتند از: (۱) آموزش از طریق تعامل با محیط، (۲) الگوریتم غیرنظارتی، (۳) نیاز به دانش قبلی دربارهٔ محیط و (٤) در حال یادگیری و خبره شدن پیوسته. این ویژگی‌ها باعث شده است که در

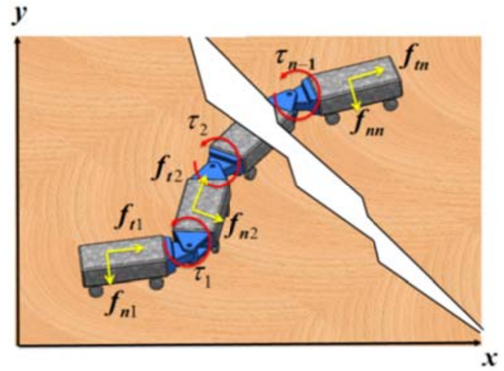
مطالعهٔ حاضر از این روش یادگیری استفاده شود. یکی از اهداف اصلی برای بررسی رباتهای متحرک، مانند ربات مارمانند، پیمودن مسیری دلخواه میباشد [17-22]. یادگیری Q، از زیرمجموعههای یادگیری تقویتی میباشد که توسط واتکین [23] معرفی شده است. تا کنون این روش بر روی رباتهای مختلف مانند رباتهای کاوشکر [26]، رباتهای انساننما [27]، رباتهای فوتبالیست [28]، رباتهای تعقیب خط [29] و رباتهای شکارچی [30,31] رباتهای خانگی [31] مورد استفاده قرار گرفته است. همچنین مقالات متعددی از روشهای تکاملی برای بهبود عملکرد یادگیری بهره گرفتهاند [32-34] هدف اصلی مقالهٔ حاضر، تأثیر پارامترهای منحنی سرپنوید بر روی یادگیری ربات مارمانند است.

مدل ریاضی برای دینامیک ربات مارمانند n عضوی

بررسی دینامیک یک ربات عموماً به دو دلیل اصلی انجام میشود: اول اینکه بهکمک معادلات دینامیکی حرکت، علاوه بر شبیهسازی، میتوان تأثیر ساختارهایِ کنترلیِ متفاوت را بر روی ربات مشاهده کرد. دوم اینکه، از این معادلات میتوان برای بهینهسازی ربات (بهعنوان مثال گشتاور مصرفی و یا فاصلهٔ پیموده شده) استفاده کرد. یکی از مشکلات موجود در بررسی رباتها، انتخاب نوع روش حل دینامیکی مناسب میباشد. در این بخش، معادلاتِ حرکت ربات مارمانند در حرکت سرپنتین و در حالت کلی بهکمک روش گیبس–اپل استخراج شده است.

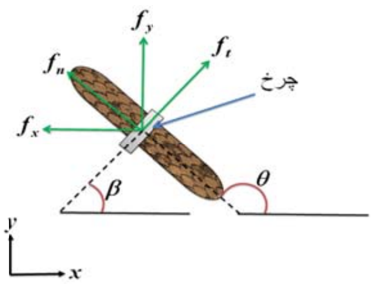
بررسی اصطکاک. با توجه بهکارهای گذشته [1,2]، عامل اصلی پیشرانش در حرکت سرپنتین، تفاوت اصطکاک در راستای عضو و عمود بر عضو است. برای ایجاد این تفاوت، در زیر عضوهای ربات چرخ قرار داده شده است (شکل ۱). مقدار نیروی اصطکاک وارد بر هر عضو توسط رابطهٔ (۱) محاسبه میشود. شایان ذکر است که در این مقاله، اصطکاک بهصورت کولمب بررسی شده است.

$$f_{ei} = -m_i C_e v_i^e \qquad (۱)$$



شکل ۱ دیاگرام آزاد ربات مارمانند بر روی سطح شیبدار

که درآن $e = t, n$ (t و n بهترتیب جهتهای مماسی و عمود بر عضو هستند). C_t و C_n ضرایب اصطکاک کولمب در جهتهای مماسی و عمودی میباشند. m_i جرم عضو i ام است.



شکل ۲ نیروی اصطکاک وارد بر هر عضو ربات مارمانند

نیروهای اصطکاک در جهتهای x و y با استفاده از معادلهٔ (۲) محاسبه میشوند. توجه داشته باشید که اگر چرخ بهصورت موازی با عضو قرار گیرد آنگاه زوایای β و θ با یکدیگر برابر خواهند بود (شکل ۲). در این شکل، θ زاویهٔ مطلق عضو و β زاویهٔ چرخ میباشد، لذا داریم:

$$f_{xi} = f_{ti}cos\beta_i - f_{ni}sin\theta_i$$
$$f_{yi} = f_{ti}\sin\beta_i - f_{ni}cos\theta_i \qquad (۲)$$

که در آن f_{xi}، f_{yi}، f_{ti} و f_{ni} به‌ترتیب اصطکاک در جهت x، y، t (مماس بر عضو) و n (عمود بر عضو) می‌باشند. در ادامه، معادلات دینامیک با استفاده از روش گیبس-اپل به‌دست می‌آید و نتایج به‌دست آمده با نرم‌افزار سیم‌مکانیک مقایسه می‌شود.

محاسبهٔ دینامیک به‌کمک روش گیبس-اپل. روش گیبس-اپل توسط گیبس در سال ۱۹۸۷ ارائه شد و در سال ۱۹۸۹ توسط اپل مورد مطالعه بیشتر قرار گرفت [24,25]. این روش فرآیند محاسبهٔ دینامیک ربات مارمانند را نسبت به روش‌های ارائه شده در مقالات [10-15] تا حد محسوسی کاهش می‌دهد. روش گیبس-اپل از جهاتی شبیه به روش لاگرانژ می‌باشد، با این تفاوت که در این روش از شتاب متغیرهای سینماتیکی استفاده می‌کند، اما در روش لاگرانژ نیاز به سرعت متغیرها داریم؛ لذا در این روش ابتدا انرژی شتاب هر یک از لینک ها، S_i، محاسبه می‌شود [5,24,25]. بنابراین داریم:

$$S_i = \frac{1}{2} m a_{ci} a_{ci} + \frac{1}{2}\alpha_i \frac{\delta H_{ci}}{\delta t} + \alpha_i(\omega_i \times H_{ci}) \qquad (۳)$$

که در آن H_{ci} مومنتوم زاویه‌ای حول مرکز جرم هر لینک، α_i شتاب زاویه‌ای و a_{ci} شتاب مرکز جرم هر لینک می‌باشد. هم‌چنین H_{ci} را به‌صورت زیر تعریف می‌کنیم.

$$H_{ci} = \begin{bmatrix} I_{xx\,i} & -I_{xy\,i} & -I_{xz\,i} \\ -I_{xy\,i} & I_{yy\,i} & -I_{yz\,i} \\ -I_{xz\,i} & -I_{yz\,i} & I_{zz\,i} \end{bmatrix} \begin{bmatrix} \omega_{x\,i} \\ \omega_{y\,i} \\ \omega_{z\,i} \end{bmatrix} \qquad (٤)$$

که در آن $I_{xx\,i}$، $I_{yy\,i}$ و $I_{zz\,i}$ مومنتوم زاویه‌ای و $I_{xy\,i}$، $I_{xz\,i}$ و $I_{yz\,i}$ حاصل‌ضرب اینرسی لینک iام می‌باشند. هم‌چنین $\omega_{x\,i}$، $\omega_{y\,i}$و$\omega_{z\,i}$ به‌ترتیب سرعت زاویه‌ای لینک *i*ام حول محور x، y و z می‌باشند. با مشتق‌گیری از معادلهٔ بالا داریم:

$$\frac{\partial H_{ci}}{\partial t} = \begin{bmatrix} I_{xx\,i} & -I_{xy\,i} & -I_{xz\,i} \\ -I_{xy\,i} & I_{yy\,i} & -I_{yz\,i} \\ -I_{xz\,i} & -I_{yz\,i} & I_{zz\,i} \end{bmatrix} \begin{bmatrix} \alpha_{x\,i} \\ \alpha_{y\,i} \\ \alpha_{z\,i} \end{bmatrix} \qquad (۵)$$

که در آن $\alpha_{x\,i}$، $\alpha_{y\,i}$ و $\alpha_{z\,i}$ به‌ترتیب شتاب زاویه‌ای لینک *i*ام حول محور x، y و z می‌باشند. با معین بودن موقعیت دم ربات در صفحهٔ حرکت و هم‌چنین زوایای مطلق هر کدام از عضوها پیکره‌بندی ربات مارمانند به‌صورت یکتا در صفحه مشخص می‌شود. مختصات تعمیم‌یافته (Generalized coordinates) به‌صورت رابطهٔ زیر تعریف می‌شود.

$$q_j = [\theta_1, \theta_2, \theta_3, \ldots, \theta_n, x_b, y_b] \qquad (٦)$$

که در آن θ_n زاویهٔ عضو *n*ام و (x_b, y_b) موقعیت دم ربات می‌باشند. فرم کلی معادلهٔ گیبس-اپل به‌صورت زیر نوشته می‌شود.

$$\frac{\delta S}{\delta q_j} = Q_j \qquad (۷)$$

که در آن Q_i نشان‌دهندهٔ نیروهای تعمیم‌یافتهٔ ناپایستار و S تابع گیبس-اپل می‌باشد. با توجه به این‌که حرکت سرپنین در صفحهٔ x-y اتفاق می‌افتد، لذا می‌توان معادلهٔ (۳) را ساده نمود. بنابراین $\omega_i = \dot{\theta}_i k$ و $\alpha_i = \ddot{\theta}_i k$. لذا در این حالت داریم:

$$\frac{\partial H_{ci}}{\partial t} = (I_{zz\,i})\ddot{\theta}_i \mathbf{k} \qquad (۸)$$

از طرفی $(\omega_i \times H_{ci})$ بر α_i عمود می‌باشد، در نتیجه ضرب داخلی آن‌ها برابر صفر خواهد بود. با قرار دادن

نتایج فوق و $a_{ci} = (\ddot{x}_{ci} + \ddot{y}_{ci})$ در معادلهٔ (۳)، انرژی شتاب کل لینک‌ها در حرکت سرپنتین را می‌توان به‌صورت زیر ساده نمود.

$$S = \sum^{i=1n} S_i = \frac{1}{2} m \sum^{i=1n} (x_{ci} + y_{ci})^2 + \frac{1}{2} I \sum^{i=1n} (\theta_i)^2 \quad (۹)$$

که در آن I، $\ddot{x}_{ci}$، $\ddot{y}_{ci}$ و $\ddot{\theta}_i$ به‌ترتیب ممان اینرسی لینک حول مرکز جرم، شتاب خطی در جهت x و y و شتاب زاویه‌ای مرکز جرم لینک iام می‌باشند. در این مطالعه تمام مفصل‌ها فعال (active) می‌باشند. در نتیجه در معادلهٔ (۹) تمام جمله‌ها وابسته به شتاب هستند و نمی‌توان از جمله‌ای برخلاف [5] صرف نظر کرد. با قرار دادن معادلهٔ (۹) در معادلهٔ گیبس-اپل (معادلهٔ ۷) معادلات دینامیک حاکم بر ربات به‌دست می‌آید. از طرفی نیروهای ناپایستار موجود در سیستم عبارتند از گشتاور موتورها، نیروهای اصطکاک و نیروهای تکیه‌گاهی. نیروهای تعمیم‌یافتهٔ ناپایستار به‌صورت زیر محاسبه می‌شوند:

$$Q_{\theta j} = d_j \left(f_{yi} \cos\theta_j - f_{xi} \sin\theta_j\right) + l_j \left[\cos\theta_j \sum^{i=j+1n} \left(f_{yi}\right) - \sin\theta_j \sum^{i=j+1n} \left(f_{xi}\right)\right] + \tau_{j-1} - \tau_j \quad (۱۰)$$

$$Q_{xb} = \sum_{i=1}^{n} \left(f_{xi}\right) \quad (۱۱)$$

$$Q_{yb} = \sum^{n} \left(f_{yi}\right) \quad (۱۲)$$

که در آن f_{xi} و f_{yi} نیروی اصطکاک به‌ترتیب در جهت x و y می‌باشند. τ_i گشتاور مورد نیاز برای عضو i است. هم‌چنین Q_{xb}، Q_{yb} و $Q_{\theta j}$ به‌ترتیب نیروهای تعمیم‌یافته در جهت x_b، y_b و θ_j هستند. با قرار دادن معادلات (۱۲-۱۰) در معادلهٔ گیبس-اپل (معادلهٔ ۷)، مدل دینامیک یک ربات مارمانند صفحه‌ای با n عضو به‌صورت زیر به‌دست می‌آید.

$$\mathbf{M}(\theta)\ddot{\mathbf{q}} + \mathbf{H}(\theta, \dot{\theta}) + \mathbf{F}(\theta) = \mathbf{B\tau} \quad (۱۳)$$

که در آن $\mathbf{M}(\theta)$ ماتریس اینرسی، یک ماتریس مثبت $(n+2)\times(n+2)$ و مثبت متقارن است. $\mathbf{H}(\theta, \dot{\theta})$ یک ماتریس $(n+2)\times 1$، شامل عبارت‌های گریز از مرکز و کوریولیس است. $\mathbf{F}(\theta)$ یک ماتریس $(n+2)\times 1$ و مربوط به نیروهای اصطکاک است. B یک ماتریس ثابت $(n+2)\times(n-1)$ است. هم‌چنین $\boldsymbol{\tau}$ یک ماتریس $(n-1)\times 1$، شامل گشتاورهای مورد نیاز است. جزئیات ماتریس‌های M،H،B و F در معادلهٔ (۱۳) در پیوست الف آورده شده‌اند. معادلات به‌دست آمده در این مطالعه را می‌توان برای هر نوع حرکت دیگر صفحه‌ای مانند حرکت آکاردئونی (Concertina Locomotion) یا حرکت خزشی (Crawling Locomotion) نیز استفاده کرد.

دینامیک مستقیم. در دینامیک مستقیم، هدف به‌دست آوردن مسیر حرکت، سرعت‌ها و شتاب ابزار ربات با داشتن نیروها و یا گشتاورهای اعمالی به مفصل‌های محرک می‌باشد. معادلهٔ (۱۳) دارای $n+2$ معادله و $n+2$ مجهول می‌باشد ($\ddot{q} \in \mathbb{R}^{n+2}$). با حل این معادله، شتاب زاویه‌ای مطلق همهٔ عضوها ($\ddot{\theta} \in \mathbb{R}^{n}$) و هم‌چنین شتاب دم ربات مارمانند ($\ddot{x}_b, \ddot{y}_b$) را می‌توان به‌دست آورد. با انتگرال‌گیری از سرعت زاویه‌ای عضوها ($\dot{\theta}$)، می‌توان زاویهٔ مفصل‌ها (θ)، موقعیت (x_b, y_b) و سرعت دم ربات ($\dot{x}_b, \dot{y}_b$) را می‌توان محاسبه کرد. بنابراین با داشتن گشتاورهای مصرفی، حرکت ربات مار مانند مشخص می‌شود.

دینامیک معکوس. در مدل‌سازی دینامیک معکوس با دانستن مسیرها، سرعت‌ها و شتاب‌های مربوط به

ربات، نیروها یا گشتاورهای مربوط به مفصل‌های محرک را محاسبه می‌کنیم. در این بخش می‌خواهیم با داشتن زاویه‌های نسبی، حرکت ربات مارمانند را ایجاد کنیم. به عبارت دیگرمی‌خواهیم روشی را استفاده کنیم که به‌کمک آن بتوانیم با داشتن زوایای نسبی و مشتقاتش ($\ddot{\varphi}$ و $\dot{\varphi}$، φ)، گشتاورهای مورد نیاز و هم‌چنین مختصات دم ربات مارمانند را به‌دست آوریم [12]. همان‌طور که می‌دانیم، رابطهٔ بین زوایای نسبی و مطلق به‌صورت زیر می‌باشد:

$$\varphi_i = \theta_{i+1} - \theta_i \tag{۱۴}$$

که در آن i=1,2,…,n-1. اگر معادلهٔ (۱۴) را به‌صورت ماتریسی بنویسیم، داریم:

$$\boldsymbol{\theta} = \mathbf{E}\boldsymbol{\varphi} + \mathbf{e}\theta_1 \tag{۱۵}$$

که درآن $\boldsymbol{\varphi}$ یک بردار n بعدی،$[\varphi_1,\varphi_2,\ldots,\varphi_{n-1}]$، θ_1 زاویهٔ مطلق عضو دم ربات، E_{ij} و $\mathbf{e}$ به‌صورت زیر تعریف می‌شوند.

$$E_{ij} = \begin{cases} 1 & i > j \\ 0 & Others \end{cases} \qquad ,\mathbf{e} = [1,1,\ldots,1]^T \tag{۱۶}$$

از طرفی می‌توان معادلهٔ (۱۳) را به‌صورت دو معادلهٔ (۱۸ و ۱۷) جداسازی کرد.

$${}^{p}\mathbf{M}(\theta)\ddot{\theta} + {}^{p}\mathbf{N}(\theta)\ddot{\mathbf{r}}_b + {}^{p}\mathbf{H}(\theta,\dot{\theta}) + {}^{p}\mathbf{f}(\theta) = \mathbf{D}\boldsymbol{\tau} \tag{۱۷}$$

$${}^{q}\mathbf{M}(\theta)\ddot{\theta} + {}^{q}\mathbf{N}(\theta)\ddot{\mathbf{r}}_b + {}^{p}\mathbf{H}(\theta,\dot{\theta}) + {}^{q}\mathbf{f}(\theta) = 0 \tag{۱۸}$$

که در آنها $\mathbf{r}_b$ بردار موقعیت دم ربات مارمانند می‌باشد. با قرار دادن مشتق دوم معادلهٔ (۱۵) در معادلهٔ (۱۸)، داریم:

$$\begin{aligned} \ddot{\mathbf{r}}_b &= -{}^{q}\mathbf{N}^{-1}\left({}^{q}\boldsymbol{M}\ddot{\boldsymbol{\theta}} + {}^{q}\mathbf{H} + {}^{q}\mathbf{f}\right) \\ &= -{}^{q}\mathbf{N}^{-1}\,{}^{q}\mathbf{M}\left(\mathbf{E}\ddot{\boldsymbol{\varphi}} + \mathbf{e}\ddot{\theta}_1\right) - {}^{q}\mathbf{N}^{-1}\left({}^{q}\mathbf{H} + {}^{q}\mathbf{f}\right) \end{aligned} \tag{۱۹}$$

حال معادلهٔ (۱۹) را در معادلهٔ (۱۷) قرار می‌دهیم. داریم:

$$\begin{aligned} &\mathbf{D}\boldsymbol{\tau} + \left({}^{p}\mathbf{N}\,{}^{q}\mathbf{N}^{-1}\,{}^{q}\mathbf{M} - {}^{p}\mathbf{M}\right)\mathbf{e}\ddot{\theta}_1 \\ &= \left({}^{p}\mathbf{M} - {}^{p}\mathbf{N}\,{}^{q}\mathbf{N}^{-1}\,{}^{q}\mathbf{M}\right)\mathbf{E}\ddot{\boldsymbol{\varphi}} \\ &\quad - {}^{p}\mathbf{N}\,{}^{q}\mathbf{N}^{-1}\left({}^{q}\mathbf{H} + {}^{q}\mathbf{f}\right) + {}^{p}\mathbf{H} + {}^{p}\mathbf{f} \end{aligned} \tag{۲۰}$$

معادلهٔ (۲۰) دارای n معادله و n مجهول ($\ddot{\boldsymbol{\theta}}_1 \in \mathbb{R}$ و $\boldsymbol{\tau} \in \mathbb{R}^{n-1}$) می‌باشد. با حل معادلهٔ (۲۰)، گشتاور مفصل‌ها (τ_i)، شتاب زاویه‌ای عضو اول $\ddot{\theta}_1$ را می‌توان محاسبه کرد. با قرار دادن این نتایج در معادلهٔ (۱۹)، می‌توانیم شتاب عضو دم ربات مارمانند یعنی $\ddot{\mathbf{r}}_b$ را به‌دست آوریم. سپس با داشتن این مقادیر می‌توانیم سرعت زاویه‌ای عضوها ($\dot{\boldsymbol{\theta}}$)، زاویهٔ مفصل‌ها ($\boldsymbol{\theta}$)، موقعیت (x_b, y_b) و سرعت دم ربات مارمانند $(\dot{x}_b, \dot{y}_b)$ را محاسبه کنیم.

یادگیری تقویتی

یکی از بهترین روش‌ها، برای یاد دادن و آموزش ربات استفاده از یادگیری تقویتی است [26-30]. در این روش حالت‌های مختلفی برای ربات تعریف می‌شود ($S = \{s_1, s_2, \ldots, s_n, \ldots\}$). در هر حالت ($s_t$)، ربات یک یا چند عمل ($a_t$) می‌تواند انجام دهد. به ازای این‌که ربات از یک حالت به کدام حالت بعدی (s_{t+1}) برود یا کدام عمل بعدی را انتخاب کند مقداری پاداش یا جریمه (r_{t+1}) می‌گیرد. هدفِ یادگیری تقویتی ایجاد نگاشتی بین حالت‌ها و عمل‌های ممکن می‌باشد؛ به عبارت دیگر، عامل (ربات) یک استراتژی $\boldsymbol{\pi}$ مناسبی را فرا می‌گیرد. لذا بایستی جمع پاداش دریافتی مورد انتظار به ماکزیمم برسد [31]. بنابراین

$$\begin{aligned} V_{(s)}^{\pi} &= E\left\{r_{(t+1)} + \gamma r_{(t+2)} + \ldots \mid s_t = s, \pi\right\} \\ &= E\left\{r_{(t+1)} + \gamma V_{s_{(t+1)}}^{\pi} \mid s_t = s, \pi\right\} \\ &= \sum_{a \in A_s} \pi(s, a)\left[r_s^a + \gamma \sum_{s'} p_{ss'}^a V_{(s')}^{\pi}\right] \end{aligned} \tag{۲۱}$$

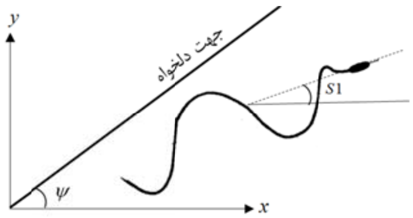
که در آن $\gamma \in [0,1]$ ضریب کاهش (Discount factor)، $\pi(s,a)$ استراتژی و احتمال انتخاب عمل a در حالت s، $p_{ss'}^{a} = Pr\{s_{t+1} = s' | s_t = s, a_t = a\}$ احتمال انتقال حالت (خاصیت ماکروف (Markov property)) و $r_s^a = E\{r_{t+1} | s_t = s, a_t = a\}$ پاداش مورد انتظار می‌باشد. هم‌چنین $V_{(s)}$ تابع ارزش حالت (Value function)s می‌باشد. روش‌های مختلفی برای به‌روزرسانی تابع ارزش وجود دارد که می‌توان به برنامه‌ریزی پویا (Dynamic programming)، مونت کارلو (Monte Carlo) و تفاضل سریع (temporal difference) (TD) اشاره نمود. در این پژوهش، از الگوریتم‌های تفاضل سریع با توجه به ویژگی‌هایشان نسبت به دو روش دیگر استفاده شده است [31]. یکی از رایج‌ترین الگوریتم‌های تفاضل سریع یادگیری تقویتی، روش یادگیری Q می‌باشد [23]. در این روش تابع ارزش عمل (action-value function) به‌صورت زیر به‌روزرسانی می‌گردد.

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \beta\left[r_{t+1} + \gamma max_a Q(s_{t+1},a) - Q(s_t,a_t)\right] \qquad (۲۲)$$

که در آن a_t عمل انتخاب شده از مجموعه عمل‌های ممکن پس از حس محیط، s_t متغیر بیان‌کنندهٔ حالت محیط، $Q(s_t,a_t)$ تابع ارزش حالت، r_{t+1} پاداش لحظه‌ای، β ضریب یادگیری و γ ضریب کاهش می‌باشند [16]. در این مقاله از روش حریص منصفانه (ε -greedy method) برای انتخاب عمل استفاده شده است؛ یعنی،

$$a_t = \begin{cases} a_t^* = \text{argmax} Q_t(a) & \text{with probability } 1-\varepsilon \\ \text{random action} & \text{with probability } \varepsilon \end{cases} \qquad (۲۳)$$

المانهای اصلی در یادگیری تقویتی برای ربات مارمانند. روش‌های یادگیری تقویتی براساس سعی و خطا می‌باشد. بنابراین برای یادگیری از ربات شبیه‌سازی شده استفاده می‌شود. سپس نتایج به‌دست آمده را بر روی یک ربات آزمایشگاهی پیاده‌سازی می‌کنیم. در این مقاله از فرمول‌های دینامیکی برای شبیه‌سازی استفاده می‌شود. سپس برای تصدیق نتایج به‌دست آمده از نرم‌افزارهای سیم مکانیک و وباتس استفاده می‌شود. در این مقاله پیش‌بینی شده که ربات مارمانند موازی با مسیری دل‌خواه حرکت کند (شکل ۳).



شکل ۳ شماتیک جهت ربات مارمانند و زاویهٔ مطلوب

قبل از اعمال الگوریتم یادگیری Q جهت ربات مارمانند موازی با محور x ها می‌باشد. برای تشویق ربات به حرکت در راستای مورد نظر، پاداش لحظه‌ای به‌صورت زیر تعریف شده است.

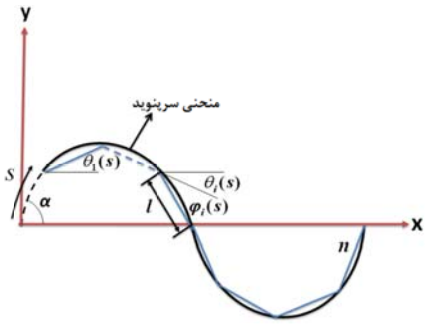
$$r = \begin{cases} S1 - \psi & if \quad S1 \le \psi \\ \psi - S1 & if \quad S1 > \psi \end{cases} \qquad (۲٤)$$

که در آن ψ زاویهٔ هدف و $S1$ جهت ربات مارمانند در لحظهٔ مورد نظر را نشان می‌دهد و به‌صورت زیر تعریف می‌شود.

$$S1 = arctan\left(\frac{y_{Head} - y_G}{x_{Head} - x_G}\right) \qquad (۲۵)$$

که در آن (x_{Head}, y_{Head}) و (x_G, y_G) به‌ترتیب موقعیت سر و مرکز ربات را نشان می‌دهند. همان‌طور

که از فرمول (۲۵) واضح است، در این مقاله باید ربات فقط در جهت مسیر مطلوب قرار بگیرد و در نتیجه فاصلهٔ بین ربات و مسیر مطلوب اهمیت ندارد. برای تولید حرکت ربات مارمانند منحنی سرپنوید استفاده شده است [1]. شکل (٤) شماتیکی از ربات مارمانند است که شامل n لینک به طول l است که توسط n-1 مفصل به یکدیگر متصل شده‌اند.



شکل ٤ قرار گرفتن عضوها بر روی منحنی سرپنوید

با استفاده از منحنی سرپنوید، زوایای نسبی به‌صورت زیر تعریف می‌شوند.

$$\varphi_i(s) = -2\alpha \sin(\frac{K_n\pi}{L}) \sin(\frac{2K_n\pi s}{L} + \frac{2K_n\pi i}{n} - \frac{K_n\pi}{n}) + \Omega \qquad (٢٦)$$

که در آن l طول هر عضو، K_n تعداد نوسانات (یا تعداد موج‌های منحنی) بدن مار، α زاویهٔ پیچشِ اولیهٔ منحنی و s طول در جهت منحنی بدن مار است. در شکل (٤)، $K_n = 1$ است [10-15]. در فرمول (٢٦) اگر Ω غیر صفر باشد، ربات مارمانند در طول یک مسیر انحنا دار حرکت می‌کند. علامت Ω ساعتگرد و یا پادساعتگرد بودن حرکت ربات مارمانند را تعیین می‌کند. در این مقاله پارامتر Ω به‌عنوان عمل (Action) برای یادگیری انتخاب شده است.

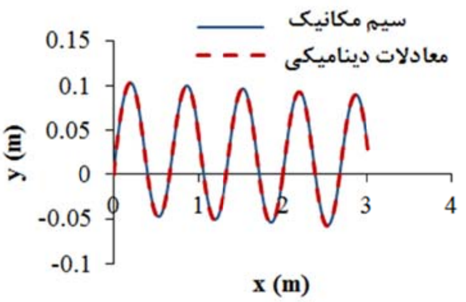
بحث و بررسی نتایج شبیه‌سازی

جدول (۱) پارامترهای شبیه‌سازی برای تصدیق معادلات دینامیکی را نشان می‌دهد. در این مقاله اصطکاک کولمب فرض شده است.

جدول ۱ پارامترهای شبیه‌سازی

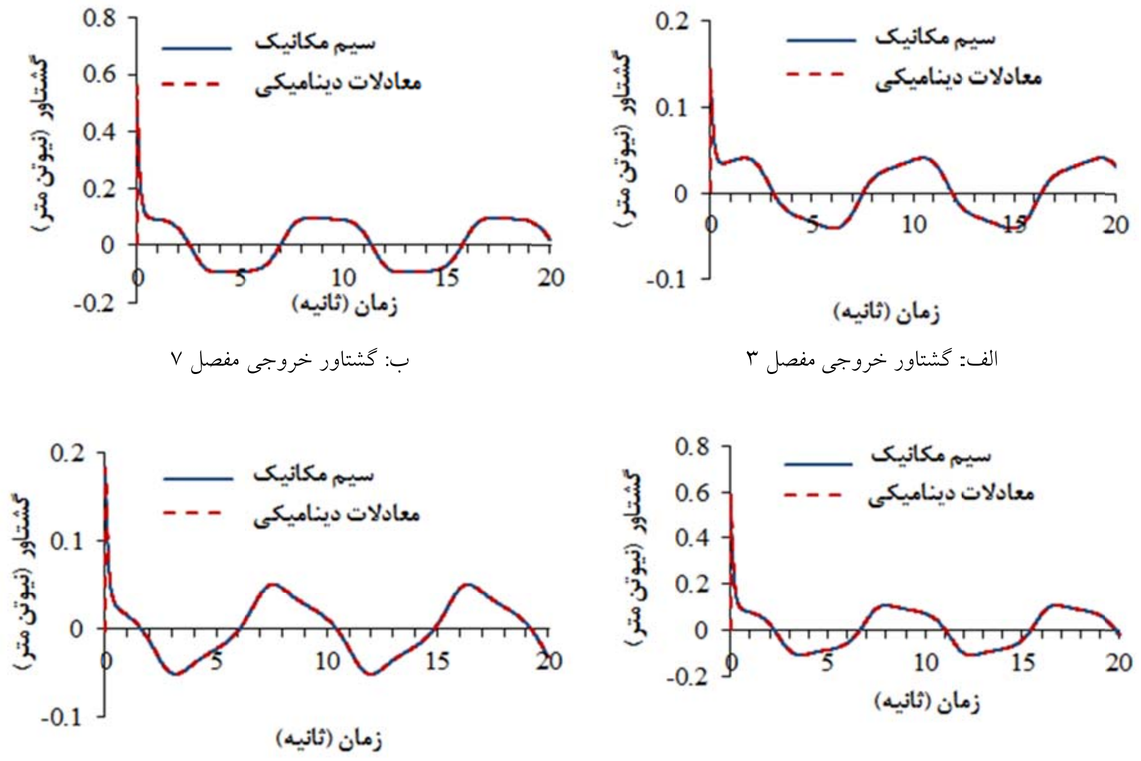
تعداد عضو	n=8, 12, 16
جرم عضو	m=0.01 kg
ضریب اصطکاک عمودی	C_n=5.5
ضریب اصطکاک مماسی	C_t=0.1
زاویهٔ پیچش اولیه	$\alpha=\pi/6$
تعداد نوسانات	$K_n = 1, 2, 3$
طول عضو	$l = 0.11, 0.15, 002$

در این قسمت به‌منظور بررسی صحت حل معادلات دینامیکی، از نرم‌افزار سیم‌مکانیک (پیوست ب) نیز استفاده شده‌است. مسیر پیموده شده توسط دم ربات، حاصل از حل معادلهٔ دینامیکی و هم‌چنین نرم‌افزار سیم‌مکانیک در شکل (۵) با یکدیگر مقایسه شده‌اند (در این‌جا $l = 0.11, K_n = 2, n = 16$).



شکل ۵ مسیر پیموده شدهٔ دم ربات

هم‌چنین نتایج حاصل از حل معادلات دینامیکی و نرم‌افزار برای گشتاورهای مورد نیاز در مفصل‌های ۳، ۷، ۹ و ۱۳ در شکل (٦) نشان داده شده‌است. شکل‌ها نشان‌دهندهٔ این مطلب هستند که نتایج حاصل از نرم‌افزار سیم‌مکانیک، نتایج حاصل از حل معادلات دینامیکی را تأیید می‌کنند.



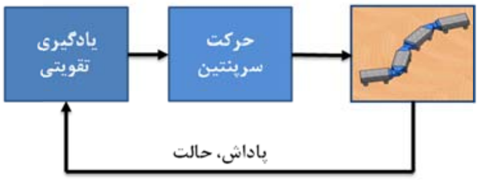
الف: گشتاور خروجی مفصل ۳
ب: گشتاور خروجی مفصل ۷
ج: گشتاور خروجی مفصل ۹
د: گشتاور خروجی مفصل ۱۳

شکل ٦ گشتاور خروجی در مفصل‌های ۳، ۷، ۹ و ۱۳

همان‌طور که در شکل (٦) پیداست، خطای بین منحنی‌های تئوری و سیم‌مکانیک بسیار ناچیز می‌باشد. علت این امر آنست که در هر دو مدل نرم‌افزار سیم‌مکانیک و تئوری از یک مدل اصطکاک (معادلهٔ ۲) استفاده شده است. با توجه به شکل (٦) می‌توان دید که گشتاور مفصل‌ها به‌صورت تناوبی در حال تغییر هستند. مقدار گشتاور در مفصل‌های ۳ و ۱۳ نسبت به گشتاور مفصل‌های ۷ و ۹ دارای مقادیر کمتری هستند. دلیل اختلاف اندازهٔ گشتاور در میان این مفاصل، فاصلهٔ هرکدام از آن‌ها تا مرکزِ جرمِ کلیِ ربات است. به بیان دیگر مفصل‌هایی که به مرکز جرم کلی ربات مارمانند نزدیک‌ترند، گشتاور بیشتری تحمل می‌کنند. هم‌چنین با مقایسه میان گشتاور مفصل‌های قرینه، که به یک نسبت از مرکز جرم فاصله دارند، دیده می‌شود که این گشتاورها از حیث دامنه تقریباً یکسانند؛ به‌عنوان مثال مفصل‌های ۳ و۱۳ نسبت به مرکز جرم کلی ربات قرینه‌اند. گشتاورهای نشان داده‌شده در شکل (٦)، در ابتدا با یک نزول آنی مواجه‌اند و سپس به‌صورت تناوبی تغییر می‌کنند. پیش از شروع حرکت سرپنتین، عضوهای ربات به‌صورت مستقیم و در راستای یکدیگر قرار دارند. با شروع حرکت، عضوها بلافاصله نسبت به یکدیگر آرایش می‌گیرند تا منحنی سرپنوید را ایجاد نمایند. این تغییر آرایش سریع، نیازمند گشتاور زیادی است. به همین دلیل در شکل (٦) گشتاور مفصل‌ها از مقدار زیادتری شروع می‌شوند.

شبیه‌سازی یادگیری Q. در این مقاله از یادگیری Q برای جهت‌یابی ربات مارمانند استفاده شده است.

شکل (۷) نمـای کلـی از چگـونگی ارتبـاط بـین ایـن الگوریتم و ربات مارمانند را نشان می‌دهد.



شکل ۷ نمای کلی از ربات مارمانند و روش یادگیری تقویتی

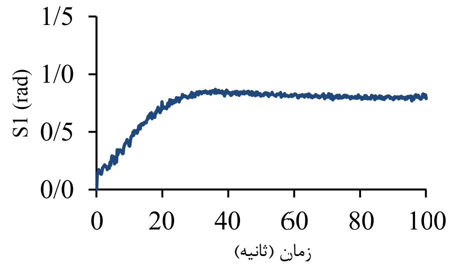
در این مقاله حالت و پاداش بـه‌ترتیـب بـا روابـط (۲۲ و ۲۱) مشخص مـی‌شـوند. شـکل (۸)، الگـوریتم یادگیری Q استفاده شده در این مقاله را نشان می‌دهـد. همان‌طور کـه مـی‌دانـیم یکـی از مشـکلات یـادگیری سیستم‌های پیچیده، کاهش نرخ یـادگیری (Slowness Learning) در طول زمان است. این مشکل را می‌تـوان با انتخاب مناسب پارامترهای یادگیری، کـاهش داد. در مقالهٔ حاضر مقادیر پارامترهای یادگیری براساس دانـش پیشین دربارهٔ تأثیر هر یـک از ایـن پارامترهـا بـر روی سرعت یادگیری، انتخاب شده‌اند. بنابراین دلیلی نـدارد که مقادیر انتخاب شده برای پارامترهای یادگیری، بهینه باشند، اما همان‌طور که قبلاً اشاره شد، هدف اصلی این مقاله بهینه کردن الگوریتم یادگیری نیست، بلکه بررسی تأثیر پارامترهـای منحنـی سـرپنوید بـر روی یـادگیری است.

هدف از این شبیه‌سازی طراحی کنترلی برای ربات مارمانند است که در آن ربات قادر به حرکت در جهت مورد نظر باشد. بایستی یادآور شویم که ۱۵ بار (تعـداد اپیزود) الگوریتم یادگیری تکرار شـده اسـت. بنـابراین نتایج نشـان داده شـده میـانگینی از ایـن تعـداد تکـرار می‌باشد. در این مقالـه جهـت مطلـوب $\psi = \frac{\pi}{4}$ اسـت. شکل‌های (۱۱-۹) به‌ترتیـب مسـیر حرکـت دم ربـات، تغییر زاویهٔ S1 و هم‌چنین پاداش دریافت شده را نشان می‌دهند. همان‌طور که در شکل‌های (۱۰ و ۹) پیداست ربات پس از گذشت زمانی موازی بـا جهـت مطلـوب $(\psi = \frac{\pi}{4} \approx 0.78)$ حرکت می‌کند. هم‌چنین شکل (۱۱) نشان می‌دهـد کـه تقریبـاً در اپیزودهـای بیشـتر از ۱۰، پاداش دریافت شده تغییر محسوسی ندارد. بـه عبـارت دیگر جهت ربات تغییر نکرده است و موازی با جهـت مطلوب می‌باشد.

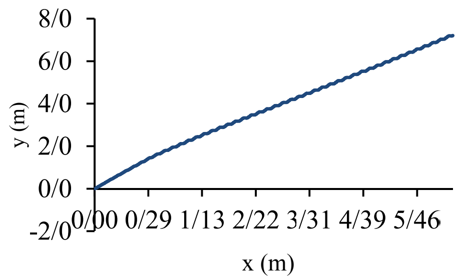
شکل ۸ الگوریتم یادگیری Q در ربات مارمانند

Set parameter $\gamma = 0.1$ and $\beta = 0.1$, and environment reward matrix R
Initialize Q(S, A) (S, is the sate set and A is action set)
For each episode:
Initialize state s
Do while not reached goal state. (Here approach to ψ, see **Fig. 3**)
Select one among all possible actions,Ω, for the current state, **S1**
Choose action$a$ form **A**
Get maximum Q value of this next state based on all possible actions.
Apply action a and observe r_{t+1}, **Eq. 24**, and s_{t+1}, **Eq. 25**
Update
Equation 22
$s \leftarrow s_{t+1}$

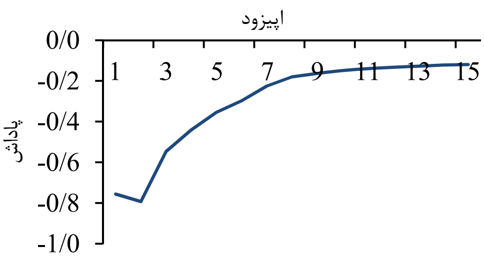
Set the next state as the current state
End Do
End For



شکل ۹ تغییرات جهت ربات در طول زمان



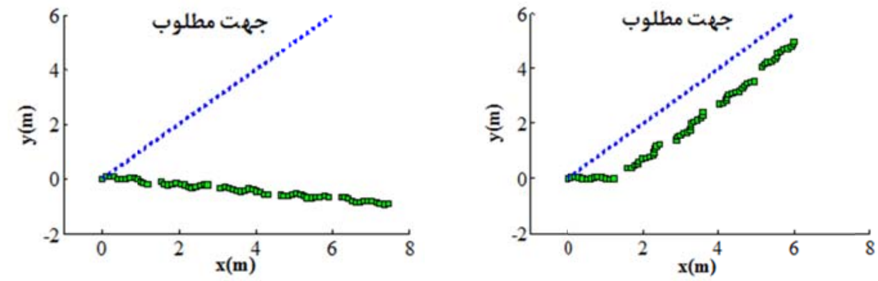
شکل ۱۰ مسیر حرکت دم ربات با وجود یادگیری Q



شکل ۱۱ پاداش دریافتی توسط ربات مارمانند در اپیزودهای مختلف

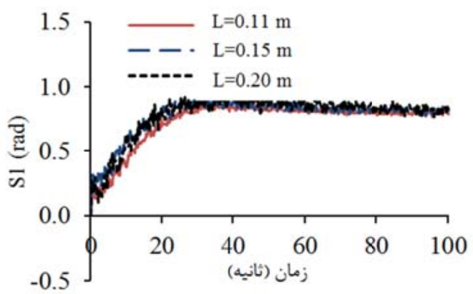
بایستی متذکر شویم در این مقاله تنها هدف این است که ربات در جهت مسیر مطلوب قرار بگیرد و در نتیجه فاصله بین ربات و مسیر مطلوب در نظر گرفته نشده است. شکل (۱۲) ربات مارمانند را قبل و بعد از یادگیری نشان می‌دهد.

تأثیر پارامترهای منحنی سرپنوید بر روی یادگیری. مارها انحناء منحنی سرپنوید را با توجه به محیطی که در آن قرار گرفته‌اند تغییر می‌دهند. لذا در این قسمت با توجه به معادلهٔ (۲۱) تأثیر تعداد نوسانات (K_n) و طول هر عضو ربات (l) و تعداد عضو (n) بر روی سرعت یادگیری بررسی شده است. بایستی متذکر شویم، از آن‌جایی که معادلات دینامیکی برای حرکت سرپنتین به‌دست آمده، لذا نتایج به‌دست آمده در این قسمت ممکن است برای همهٔ ربات‌های مارمانند صادق نباشد. شکل (۱۳) جهت ربات مارمانند را در طول زمان با تغییر پارامترهای منحنی سرپنوید نشان می‌دهد. همان‌طور که در شکل (۱۳-الف و ب) پیداست پارامترهای طول عضو و تعداد عضو تأثیر محسوسی بر پارامتر S1 ندارند. در صورتی که پارامتر K_n تأثیر محسوسی بر این پارامتردارد (شکل ۱۳-ج).

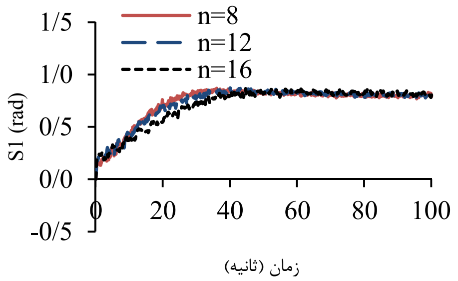


الف) بعد از آموزش ب) قبل از آموزش

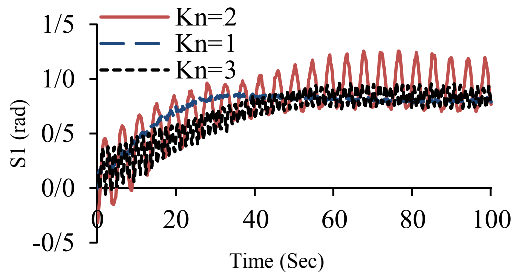
شکل ۱۲ ربات مارمانند



الف) $K_n = 2$، $n = 8$ و $\alpha = \frac{\pi}{6}$



ب) $K_n = 2$، $l = 0.11m$ و $\alpha = \frac{\pi}{6}$



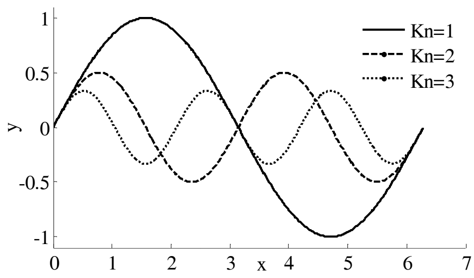
ج) $n = 8$، $l = 0.11m$ و $\alpha = \frac{\pi}{6}$

شکل ۱۳ تغییرات جهت ربات مارمانند در طول زمان

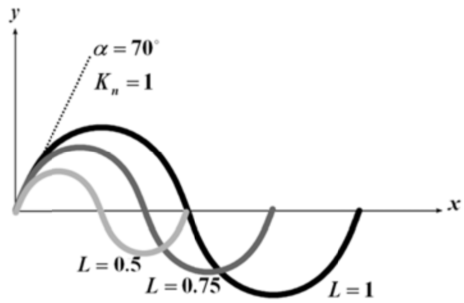
علت تأثیر پارامتر K_n را می‌توان در این دانست که این پارامتر شکل ربات را تغییر می‌دهد. براساس شکل (۱٤) تغییر در تعداد موج (K_n) تنها بر ارتفاع منحنی تأثیرگذار است و تأثیری بر عرض منحنی ندارد. اما پارامترهای فیزیکی- پارامترهای l و n - تنها طول ربات را تغییر می‌دهند و بر روی شکل ربات تأثیری ندارد. با توجه به شکل (۱۵) تغییر $L = nl$ اندازهٔ منحنی را زیاد یا کم می‌کند و بر شکل منحنی تأثیر نمی‌گذارد.

شکل (۱٦) مسیر پیموده شده توسط دم ربات با تغییر پارامترهای منحنی سرپنوید را نشان می‌دهد. شکل‌های (۱٦-الف تا ج) به‌ترتیب نشان می‌دهند که ربات در $l = 0.11\text{m}$، $n = 8$ و $K_n = 2$ عملکرد بهتری داشته است. در نتیجه با انتخاب پارامترهای $\alpha = \frac{\pi}{6}$، $K_n = 2$، $l = 0.11\text{m}$ و $n = 8$ (خط پر) ربات مارمانند سریع‌تر در جهت مطلوب (یعنی در جهت فلش) قرار می‌گیرد. به‌علاوه می‌توان نتیجه گرفت که K_n بیشترین تأثیر را بر یادگیری ربات می‌گذارد. بایستی خاطر نشان کنیم که گرچه این نتیجه کلی نمی‌باشد، اما بررسی تعامل پارامترهای منحنی سرپنوید ضروری به‌نظر می‌رسد. در ادامه قصد داریم تعامل این پارامترها را با الگوریتم‌های تکاملی بررسی نماییم.

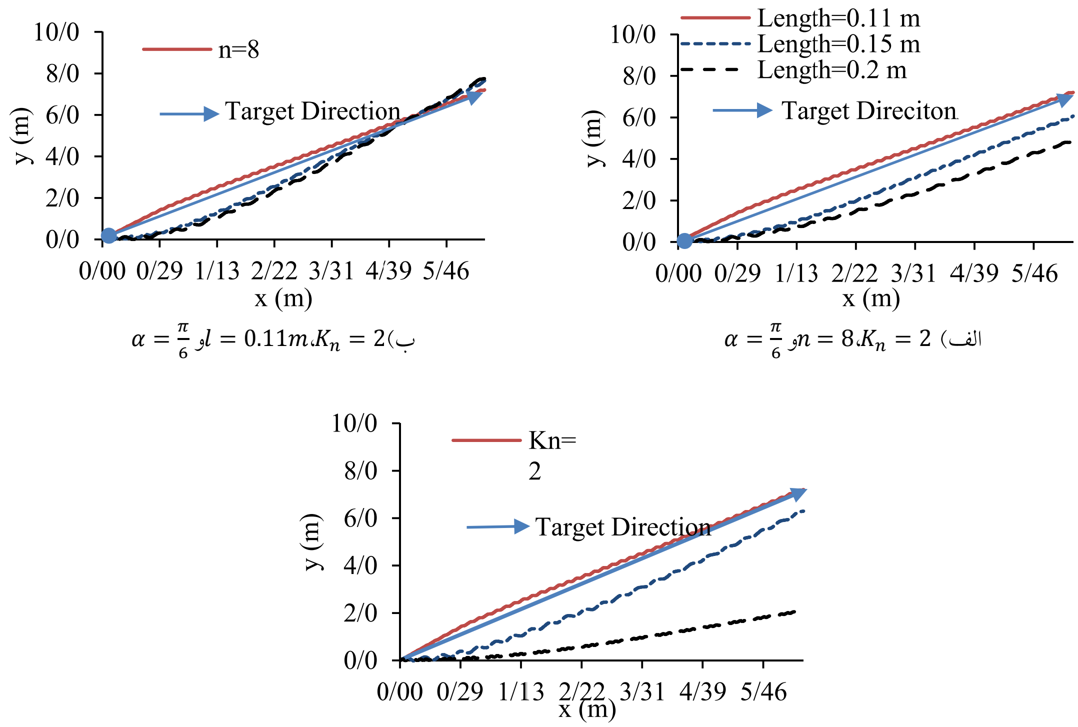
شکل (۱۷) میانگین پاداش دریافتی توسط ربات مارمانند را در اپیزودهای مختلف با تغییر پارامترهای منحنی سرپنوید نشان می‌دهد. از آن‌جایی که پارامتر K_n بیشترین تأثیر را بر شکل ربات مارمانند دارد، می‌توان انتظار داشت که تأثیر بیشتری بر سرعت یادگیری بگذارد. شکل (۱۷-ج) نشان می‌دهد که در $K_n = 2$ پاداش دریافت شده در انتها تقریباً ثابت است در حالی که در 3 و $K_n = 1$ این‌طور نیست. این نشان می‌دهد که در این مقادیر ربات بعد از مدت زمان شبیه‌سازی، هنوز کاملاً موازی با جهت مطلوب نشده است که خود حاکی از کاهش سرعت یادگیری در این مقادیر از K_n می‌باشد.



شکل ۱٤ تأثیر زاویهٔ پیچشی اولیه و تعداد موج بر منحنی سرپنوید [15]



شکل ۱۵ منحنی سرپنوید در مختصات کارتزین و تأثیر L بر این منحنی [15]



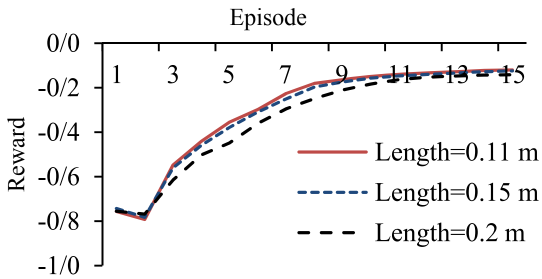
الف) $\alpha = \frac{\pi}{6}$ و $n = 8$،$K_n = 2$

ب) $\alpha = \frac{\pi}{6}$ و $l = 0.11m$،$K_n = 2$

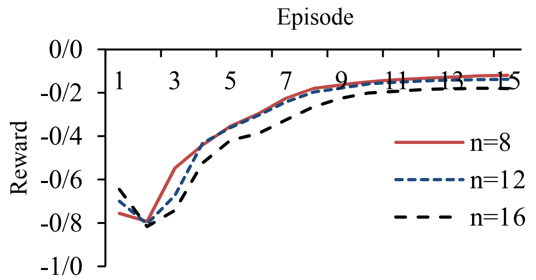
ج) $\alpha = \frac{\pi}{6}$ و $l = 0.11m$،$n = 8$

شکل ۱٦ مسیر پیموده شده توسط دم ربات

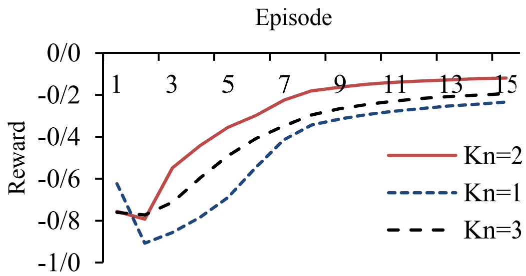
(فلش نشان‌دهندهٔ جهت مطلوب است)



الف) $\alpha = \frac{\pi}{6}$ و $n = 8$، $K_n = 2$



ب) $\alpha = \frac{\pi}{6}$ و $l = 0.11m$، $K_n = 2$

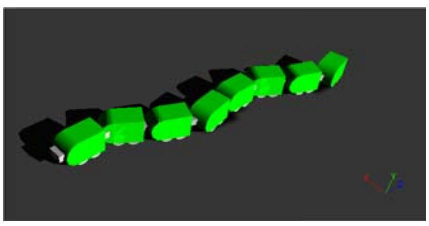


ج) $\alpha = \frac{\pi}{6}$ و $l = 0.11m$، $n = 8$

شکل ۱۷ میانگین پاداش دریافتی توسط ربات مارمانند در اپیزودهای مختلف

نرم‌افزار وباتس

در این قسمت زوایای به‌دست آمده از فرآیند یادگیری را به ربات مارمانند شبیه‌سازی شده در نرم‌افزار وباتس اعمال خواهیم کرد. از آن‌جایی که این نرم‌افزار به مانند محیط واقعی می‌باشد، لذا رباتی که دراین نرم‌افزار شبیه‌سازی شود و با موفقیت مأموریتش را انجام بدهد در محیط طبیعی و پس از ساخت هم با همین موفقیت روبه‌رو خواهد شد. وباتس یک محک بسیار عالی برای ساخت یک ربات است [12,15]. شکل (۱۸) نمایی از مدل ربات مارمانند را در این نرم‌افزار نشان می‌دهد.

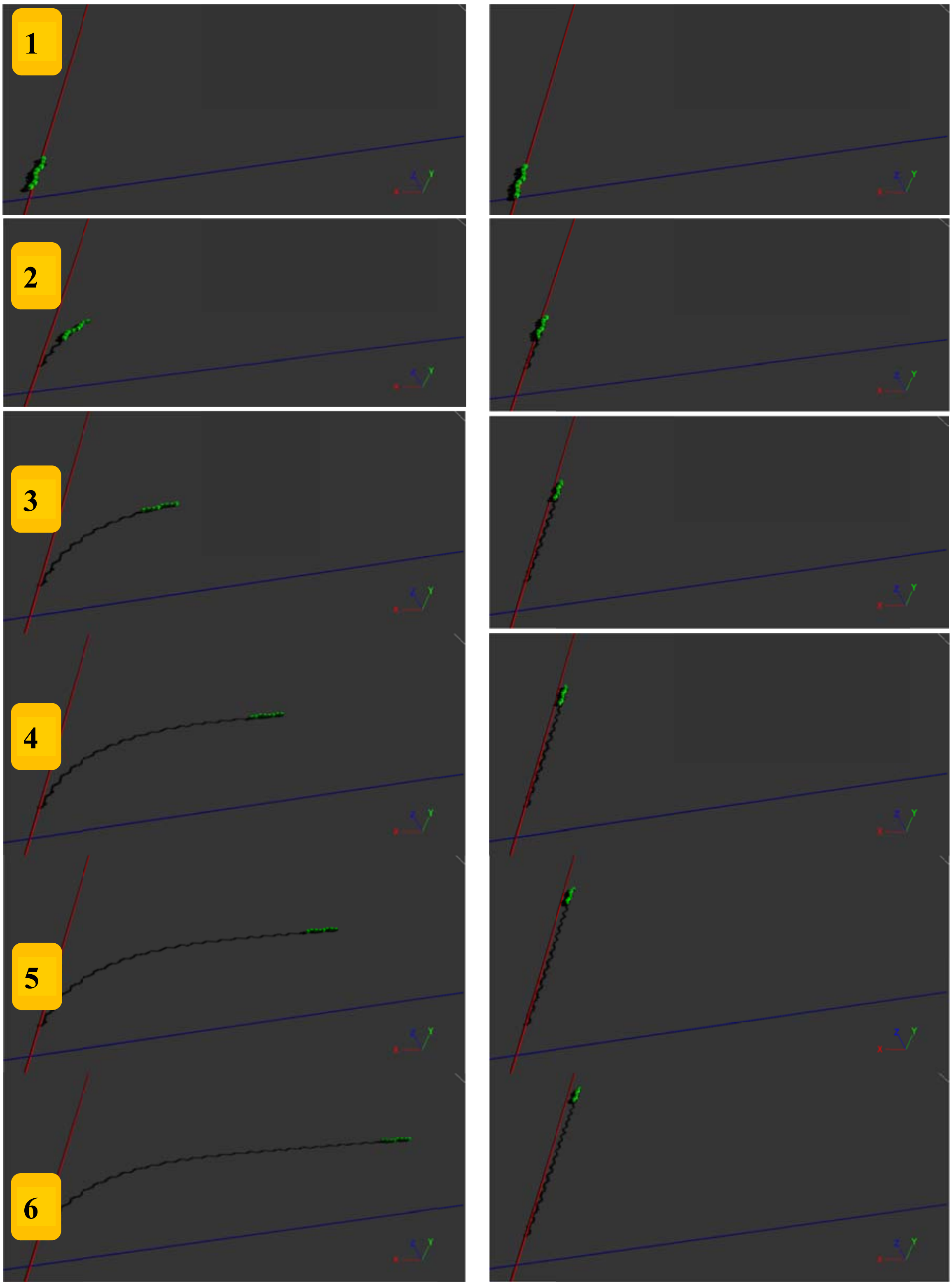


شکل ۱۸ نمایی از ربات مارمانند با ۸ عضو در نرم‌افزار وباتس

شکل (۱۹-الف) مسیر حرکت ربات مارمانند را قبل از آموزش نشان می‌دهد. همان‌طور که در شکل (۱۹-ب) واضح می‌باشد، ربات مارمانند پس از یادگیری، موازی با مسیر مورد نظر حرکت می‌کند.

ربات مارمانند FUM- Snake II

ربات نشان داده شده در شکل (۲۰) با نام FUM-Snake II از نسل ربات‌های مارمانند است که در گروه رباتیک دانشگاه فردوسی مشهد طراحی و ساخته شده است. برای مدل‌سازی حرکت سرپنتین در زیر هر عضو، چهار چرخ تعبیه گشته است. طول هر عضو ربات 110 mm و وزن هر عضو 90 g است. هر عضو دارای عرض و ارتفاعی به‌ترتیب 60 mm و 50 mm است. شکل (۲۰) نحوهٔ ارتباط ربات با محیط اطراف را نشان می‌دهد. جدول (۲) مقایسهٔ این ربات با ربات‌های مارمانند مشهور را نشان می‌دهد.



ب)ربات مارمانند بعد از یادگیری

الف) ربات مارمانند قبل از یادگیری

شکل ۱۹ ربات مارمانند شبیه‌سازی شده در نرم‌افزار وباتس

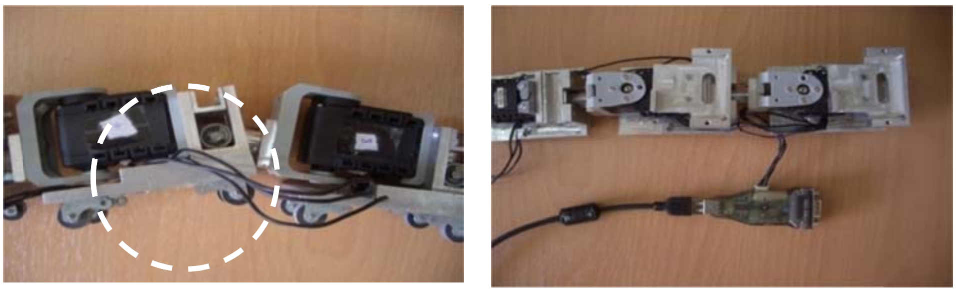


شکل ۲۰ ربات مارمانند FUM- Snake II و اتصالات

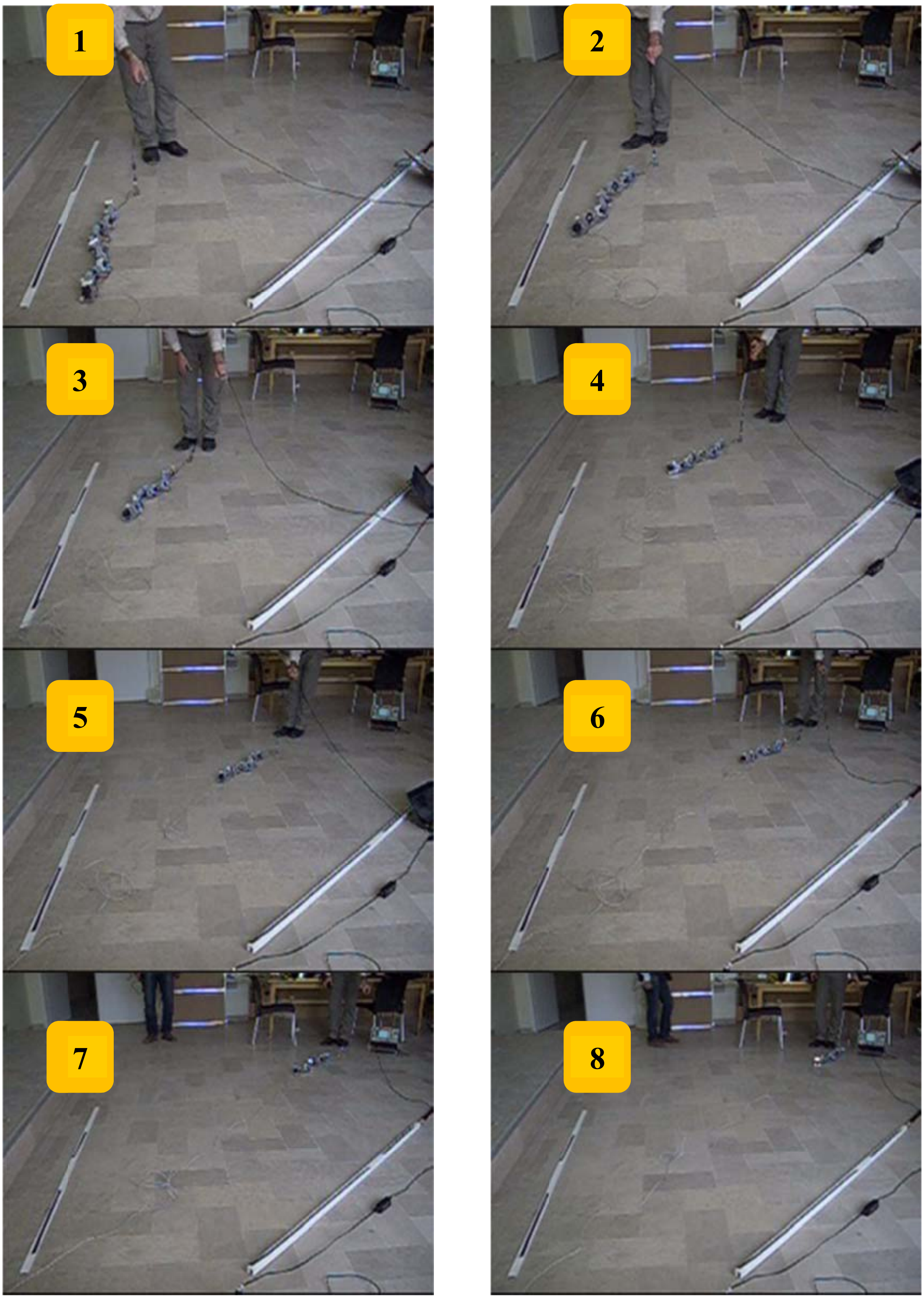
جدول ۲ مقایسهٔ ربات مارمانند FUM- Snake II با ربات‌های معروف [12]

نام ربات	نوع	طول کل (mm)	سطح مقطع (m^2)	وزن کل (kg)	سرعت ماکزیمم (mm/s)	تعداد ماژول
ACM III	I	2000	0.023	28	400	20
AmphiBot I (AB I)	I	490	0.002	---	35	8
AmphiBot II (AB II)	I	770	0.002	---	400	8
FUM Snake-2	**I**	**880**	**0.003**	**2.5**	**350**	**8**
KR-II	II	3300	0.497	370	500	7
KR-I	III	1390	0.081	27.8	266	6
OmniTread (OT-8)	III	1270	0.034	13.6	100	5
OmniTread (OT-4)	III	940	0.007	3.6	150	7
JL-I	III	1050	0.038	21	180	3
Kotay’s Inchworm I (KIR-1)	IV	250	---	0.455	4	---
Kotay’s Inchworm II (KIR-2)	IV	330	---	0.566	13	---
CMU (M1)	IV	840	0.003	1.26	102	---
FUM Snake-3	**IV**	**670**	**0.0045**	**1.4**	**105**	**9**
Planar Inchworm (PI)	V	710	---	6	1	2
Slim Slime Robot (SSR)	V	730	0.013	12	60	6

I: ربات‌های چرخ‌دار غیرفعال (robots with passive wheels) – II: ربات‌های چرخ‌دار فعال (robots with active wheels) – III: ربات‌های مجهز به پا (robots with active treads) – IV: ربات‌های موج‌شکل ویا موج‌های عمودی (robots based on undulation using vertical waves) – V: ربات‌های بر اساس حرکت موج با استفاده از انبساط‌های خطی (robots based on undulation using linear expansion)



شکل ۲۱ فاصلهٔ ایجاد شده در مفصل عضوها



شکل ۲۲ حرکت سرپنتین ربات مارمانند FUM-Snake II

همان‌طور که از جدول (۲) پیداست این ربات ویژگی‌های مناسبی (سرعت و وزن ربات) نسبت به هم گروه‌های خود دارد. از ویژگی‌های جالب دیگر این ربات طراحی مفصل‌های آن می‌باشد. برای سازگارتر ساختن ربات در محیط‌هایی با سطوح نسبتاً ناهموار، در طراحی مفاصل مقداری لقی (درجهٔ آزادی در جهت عمود بر حرکت) لحاظ شده است. این طراحی در شکل (۲۱) آمده است. از طرفی هیروس [1] نشان داد که برای ایجاد و پیش‌روی ربات مارمانند در حرکت سرپنتین بایستی ناهمسانگردی در ضریب اصطکاک بین ضرایب اصطکاک افقی و عمودی وجود داشته باشد. برای این منظور، چرخ‌های کوچکی در زیر هر عضو و در موازات با هر عضو تعبیه شده است (شکل ۲۱). این کار باعث کاهش ضریب اصطکاک در جهت مماسی و افزایش ضریب اصطکاک در جهت عمود بر لینک می‌شود.

در انتها برای صحت‌سنجی نتایج به‌دست آمده از شبیه‌سازی، این نتایج را بر روی ربات آزمایشگاهی اعمال می‌کنیم. شکل (۲۲) پیش‌روی ربات FUM-Snake II را بعد از یادگیری نشان می‌دهد. در این مقاله زوایای هر یک از مفاصل در فرآیند یادگیری از طریق شبیه‌سازی ابتدا به‌دست آمده و سپس زوایای حاصل به موتور متناظر اعمال گردیده‌اند. به عبارت دیگر ورودی ربات تنها زوایای موتورها می‌باشد و این موتورها از موقعیت ربات نسبت به مسیر سیگنال کنترلی دیگری دریافت نمی‌کنند. در این مطالعه فقط می‌خواهیم ربات در جهت مسیر مطلوب قرار بگیرد و در نتیجه فاصلهٔ بین ربات و مسیر مطلوب در نظر گرفته نشده است.

جمع‌بندی و نتیجه‌گیری

در این مقاله حرکت سرپنتین ربات مارمانند مورد مطالعه قرار گرفته است. معادلات دینامیکی حاکم بر ربات به‌کمک روش گیبس-اپل در حالت کلی استخراج شده‌اند. روش ارائه شده در این مقاله حجم محاسبات دینامیک ربات مارمانند را تا حد محسوسی کاهش می‌دهد. برای تصدیق معادلات دینامیکی از نرم‌افزار سیم‌مکانیک استفاده شده است. نتایج نشان می‌دهند که میزان پیش‌روی و گشتاور مفاصل ربات در معادلات دینامیکی و نرم‌افزار سیم‌مکانیک از تطابق بسیار خوبی برخوردار است. در این مقاله، اصطکاک به‌صورت کولمب در نظر گرفته شده است.در این مطالعه نشان داده شد، مفصل‌هایی که به مرکز جرم کلی ربات مارمانند نزدیک‌ترند، گشتاور بیشتری تحمل می‌کنند. هم‌چنین با مقایسه میان گشتاور مفصل‌های قرینه، که به یک نسبت از مرکز جرم فاصله دارند، دیده می‌شود که این گشتاورها از حیث دامنه تقریباً یکسانند.

در این مقاله برای اولین بار از روش یادگیری Q در جهت‌یابی ربات مارمانند استفاده شده است. به‌علاوه تأثیر پارامترهای منحنی سرپنوید بر سرعت یادگیری ربات با استفاده از معادلات دینامیکی از دیگر موارد مورد بررسی در این تحقیق می‌باشد. بر طبق نتایج شبیه‌سازی طول عضوها و تعداد عضوها تأثیر چندانی بر روی یادگیری ربات مارمانند ندارند زیرا این پارامترها تنها طول ربات را تغییر می‌دهد و بر روی شکل ربات تأثیری ندارند. به عبارت دیگر پارامترهای فیزیکی بر سرعت یادگیری تأثیر نمی‌گذارند.

از آن‌جایی که پارامتر K_n شکل ربات مارمانند را تغییر می‌دهد، لذا بر سرعت یادگیری تأثیر محسوسی دارد.

برای آزمایش و تصدیق نتایج حاصل از یادگیری، از نرم‌افزار وباتس و ربات مارمانند FUM-Snake II استفاده شده است. نتایج تحقیق بیانگر این می‌باشند که روش یادگیری Q برای جهت‌یابی ربات مارمانند مناسب می‌باشد.

در مطالعهٔ حاضر، به‌طور موفقیت‌آمیزی موارد زیر

بررسی شده است:

(۱) استخراج معادلات دینامیک ربات از روش گیبس- اپل.

(۲) استفاده از یادگیری Q برای جهت‌یابی ربات مارمانند.

(۳) بررسی تأثیر پارامترهای فرم بدن ربات در یادگیری.

(٤) تصدیق نتایج شبیه‌سازی با استفاده از نرم‌افزارهای سیم‌مکانیک، وباتس و ربات آزمایشگاهی.

در پژوهش‌های آینده قصد داریم تعامل پارامترهای منحنی سرپنوید را با الگوریتم های تکاملی بررسی نماییم. هم‌چنین مسیر مطلوب را انحنادار کنیم و پاداش دریافتی را طوری تعریف نماییم که گشتاور مصرفی و فاصلهٔ ربات مارمانند تا مسیر مطلوب را نیز در نظر گرفته باشد.

مراجع

1. Hirose, S., "*Biologically Inspired Robots (Snake-like Locomotor and Manipulator)*", Oxford University Press, Oxford, (1993).
2. Saito M., Fukaya M. and Iwasaki T., "Serpentine locomotion with robotic snakes", *IEEE Control Systems Magazine*, Vol. 22, pp. 64–81, (2002).
3. Transeth, A.A., "Modelling and Control of Snake Robots", Ph.D. Dissertation, Trondheim, September, (2007).
4. Transeth, A.A., Pettersen K.Y. and Liljebäck, P., "A survey on snake robot modeling and locomotion", *Robotica*, Vol. 27, pp. 999–1015, (2009).
5. Vossoughi Gh., Pendar Ho., Heidari Z. and Mohammadi S., "Assisted passive snake-like robots: conception and dynamic modeling using Gibbs–Appell method", *Robotica*, Vol. 26, pp. 267–276, (2008).
6. Spranklin, B.W., "Design, Analysis and fabrication of a snake- inspired robot with a rectilinear gait", Master of Science Thesis, University of Maryland, (2006).
7. Liljebäck, P., Stavdahl, A. and Pettersen K.Y., "Modular pneumatic snake robot: 3D modeling, implementation and control", *in: Proc. 16th IFAC World Congress*, Prague, Czech Republic, 4-8 July (2005).
8. Ye, Ch., Ma, Sh., and Li, B., "Development of a 3D Snake-like Robot Perambulator-II: Design and Basic Experiments", *in Proc. 2007 IEEE Int. Conf. on Intelligent Mechatronics and Automation (ICMA2007)*, pp. 117-122, 5-8 Aug, (2007).
9. Crespi, A. and Ijspeert, A.J., "Online Optimization of Swimming and Crawling in an Amphibious Snake Robot", *IEEE Transactions on Robotics*, Vol. 24, (2008).
10. Hasanzadeh, Sh. and Akbarzadeh, Alireza., "Ground adaptive and optimized locomotion of snake robot moving with a novel gait", *Auton Robot*, Vol. 28, pp. 457–470, (2010).
11. Hasanzadeh, Sh. and Akbarzadeh, A., "Adaptive Optimal Locomotion of Snake Robot Based on CPG-Network Using Fuzzy Logic Tuner", *IEEE - CIS RAM*, (2008).
12. Kalani, H. and Akbarzadeh, A., "Design and Modeling of a Snake Robot Based on Worm-Like Locomotion", Accepted, *Advance Robotics*.
13. Kalani, H., Akbarzadeh, A. and Safehian, J., "Traveling Wave Locomotion of Snake Robot along Symmetrical and Unsymmetrical body shapes", *ISR-Robotik Conf.*, Munich, Germany, 2-4 June (2010).
14. Safehian, J., kalani, H. and Akbarzadeh, A., "A Novel Kinematics Modeling Method for Snake Robot

in Traveling wave Locomotion ASME 2010 10th Biennial Conference on Engineering Systems Design and Analysis ", Turkish, 12-14 July (2010).

15. Kalani, H., Akbarzadeh, A. and Bahrami, Ho., "Application of Statistical Techniques in Modeling and Optimization of a Snake Robot", *Robotica*, Vol. 31, pp. 623-641, (2013).
16. Sutton, R.S., and Barto, A.G., "*Reinforcement Learning: An Introduction*". MIT Press, Cambridge, MA, (1998).
17. Chen, C., Li, H. and Dong, D., "Hybrid control for robot navigation: A hierarchical Q-learning algorithm", *IEEE Robot. Autom. Mag*., Vol. 15, pp. 37–47, (2008).
18. Franchi, A., Freda, L. and Oriolo, G., "The sensor-based random graph method for cooperative robot exploration", *IEEE/ASME Trans. Mechatronics*, Vol. 14, pp. 163–175, (2009).
19. Chen, C. and Dong, D., "Grey system based reactive navigation of mobile robots using reinforcement learning", *Int. J. Innov. Comput., Inf. Control*, Vol. 6, pp. 789–800, (2010).
20. Jan, G.E., Chang, K.Y. and Parberry, I., "Optimal path planning for mobile robot navigation", *IEEE/ASME Trans. Mechatronics*, Vol. 13, pp. 451–460, (2008).
21. Rodriguez, M., Iglesias, R., Regueiro, C.V., Correa, J. and Barro, S., "Autonomous and fast robot learning through motivation", *Robot. Auton.Syst.*, Vol. 55, pp. 735–740, (2007).
22. Ratliff, N.D., Silver, D. and Bagnell, J.A., "Learning to search: Functional gradient techniques for imitation learning", *Auton. Robots*, Vol. 27, pp. 25– 53, (2009).
23. Watkins, J.C.H. and Dayan, P., "Q-learning. Machine Learning", Vol. 8, pp. 279-292, (1992).
24. Ginsberg, J.H., "*Advanced Engineering Dynamics*", New York, Cambridge University Press, (1998).
25. Greenwood, D.T., "*Advanced Dynamics*", New York, Cambridge University Press, (2003).
26. Matari´c, M.J., "Reinforcement learning in the multi-robot domain", *Autonomous Robots*, Vol. 4, pp. 73–83, (1997).
27. Peters, J., Vijayakumar, S. and Schaal, S.," Reinforcement learning for humanoid robotics", *In Third IEEE-RAS International Conference on Humanoid Robots*, Karlsruhe, 29-30 Sept (2003).
28. Asada M., Uchibe E. and Hosoda, K., "Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development", *Artificial Intelligence*, Vol. 110, pp. 275–292, (1999).
29. Park K.H., Kim Y.J. and Kim J.H., "Modular Q-learning based multi-agent cooperation for robot soccer", *Robotics and Autonomous Systems*, Vol. 35, pp. 109–122, (2001).
30. Tan, M., "Multi-agent reinforcement learning: independent vs. cooperative agents", *in Proceedings of the Tenth International Conference on Machine Learning*, Amherst, MA, pp. 330–337, (1993).
31. Dong, D., Chen, Ch., Chu, J. and Tarn, T.J., "Robust Quantum-Inspired Reinforcement Learning for Robot Navigation", *IEEE/ASME Transactions on Mechatronics*, Vol. 17, (2012).
32. Ito, K., Fukumori, Y. and Takayama, A., "Autonomous control of real snake-like robot using reinforcement learning; abstraction of state-action space using properties of real world", *Intelligent Sensors, Sensor Networks and Information*, ISSNIP, 3rd International Conference, Melbourne, Qld., 3-6 Dec., (2007).
33. Ito, K., Kamegawa, T. and Matsuno, F., "Extended QDSEGA for controlling real robots-acquisition of locomotion patterns for snake-like robot", *Robotics and Automation*, Vol. 1, (2003).
34. Tanev, I., Ray, T. and Buller, A., "Automated evolutionary design, robustness, and adaptation of sidewinding locomotion of a simulated snake-like robot", *Robotics, IEEE Transactions*, pp. 632-645, (2005).

پیوست الف

جزئیات ماتریس‌های **M**،**H**،**B** و **F** در معادلهٔ (۱۳) برای یک ربات مارمانند n عضوی و بر روی سطح شیب‌دار در حرکت سرپنتین به‌صورت زیر می‌باشد.

$$\mathbf{M}=\begin{bmatrix} {}^{p}M_{n\times n} & {}^{p}N_{n\times 2} \\ {}^{q}M_{2\times n} & {}^{q}N_{2\times 2}\end{bmatrix},$$

$${}^{q}\mathbf{N}=\begin{bmatrix} \sum_{i=1}^{n} m_i & 0 \\ 0 & \sum_{i=1}^{n} m_i \end{bmatrix},$$

$${}^{p}\mathbf{N}={}^{q}\mathbf{M}^{\mathrm{T}},$$

$${}^{p}M_{ij}=\begin{cases}\left[m_j d_j l_j+\left(\sum_{k=j+1}^{n} m_k\right) l_i l_j\right] cos(\theta_i-\theta_j) & i<j, 1\le j\le n \\ I_i+m_i d_i^2+l_i^2\left(\sum_{j=i+1}^{n} m_j\right) & i=j, 1\le j\le n \\ {}^{p}M_{ji} & i>j, 1\le j\le n\end{cases}$$

$${}^{q}M_{2,j}=cos\theta_j\left[m_j d_j+\left(\sum_{k=j+1}^{n} m_k\right) l_j\right] for \quad (j=1,2,\dots,n)$$

$${}^{q}M_{1,j}=-sin\theta_j\left[m_j d_j+\left(\sum_{k=j+1}^{n} m_k\right) l_j\right] for \quad (j=1,2,\dots,n)$$

$$=\begin{bmatrix} {}^{p}\mathbf{H}_{n\times 1} \\ {}^{q}\mathbf{H}_{2\times 1}\end{bmatrix},$$

$${}^{p}H_i=l_i\sum_{j=i+1}^{n}\left\{\left[m_j d_j+l_j\left(\sum_{k=j+1}^{n} m_k\right)\right] sin(\theta_i-\theta_j)\dot{\theta}_j^2\right\}+\sum_{j=1}^{i-1}\left\{\left[m_i d_i+l_i\left(\sum_{k=i+1}^{n} m_k\right)\right] l_j sin(\theta_i-\theta_j)\dot{\theta}_j^2\right\} \quad (i=1,2,\dots,n)$$

$${}^{q}\mathbf{H}=\begin{bmatrix} -\sum_{i=1}^{n} cos\theta_i\left[m_i d_i+\left(\sum_{k=i+1}^{n} m_k\right) l_i\right]\dot{\theta}_i^2 \\ -\sum_{i=1}^{n} sin\theta_i\left[m_i d_i+\left(\sum_{k=i+1}^{n} m_k\right) l_i\right]\dot{\theta}_i^2\end{bmatrix}$$

$$\mathbf{F}=\begin{bmatrix} {}^{p}\mathbf{f}_{n\times 1} \\ {}^{q}\mathbf{f}_{2\times 1}\end{bmatrix}$$

$${}^{p}f_j=d_j(f_{xj}sin\theta_j-f_{yj}cos\theta_j)+l_j\left[sin\theta_j\sum_{i=j+1}^{n}(f_{xi})-cos\theta_j\sum_{i=j+1}^{n}(f_{yi})\right]$$

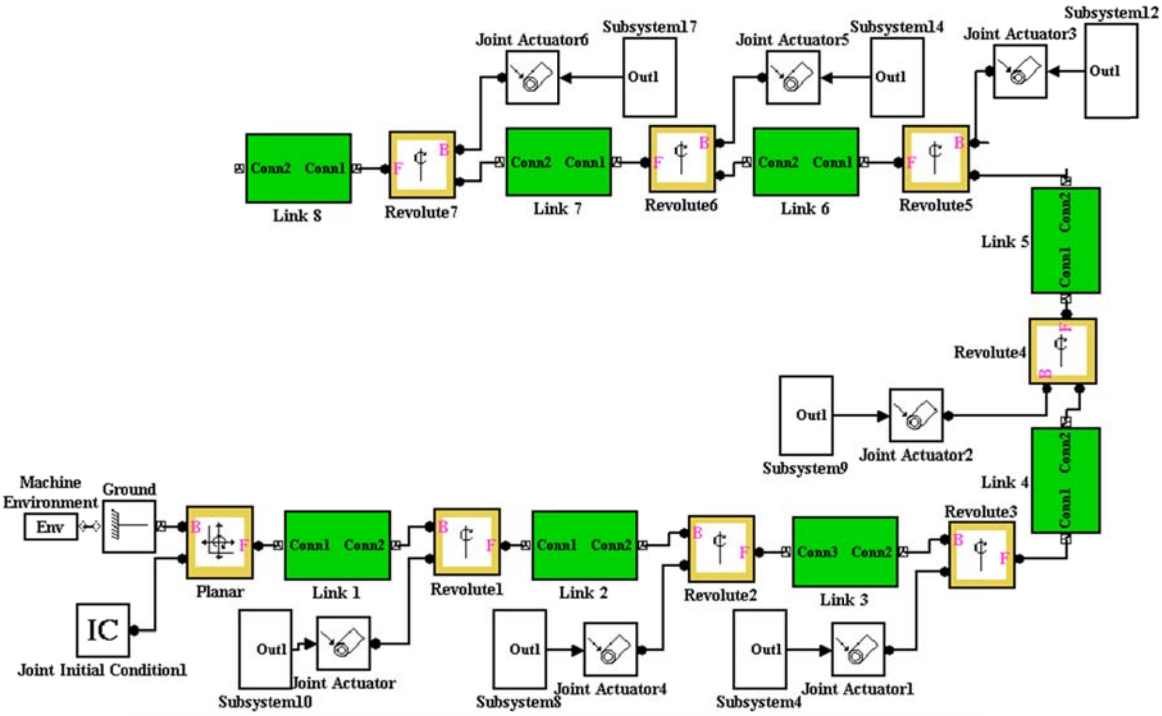
$${}^{q}\mathbf{f}=\begin{bmatrix} -\sum_{i=1}^{n}(f_{xi}) \\ -\sum_{i=1}^{n}(f_{yi})\end{bmatrix}$$

$$\mathbf{B}=\begin{bmatrix} D_{n\times n-1} \\ 0\end{bmatrix},$$

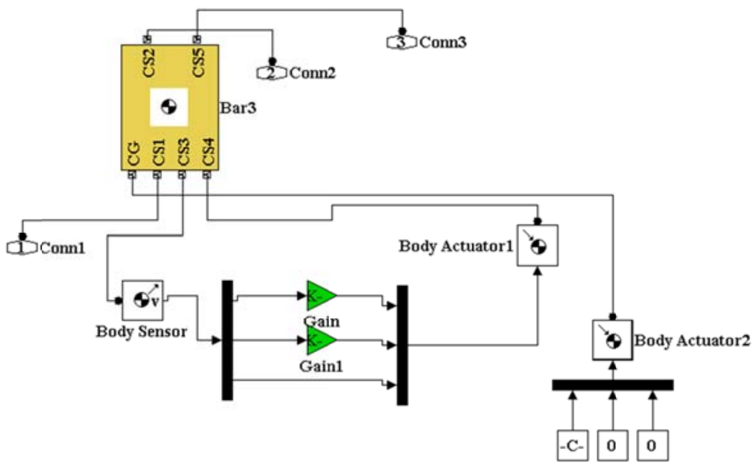
$$D_{ij}=\begin{cases}-1 & i=j \\ 1 & i=j+1 \\ 0 & Others\end{cases}$$

پیوست ب

به‌منظور بررسی صحت حل معادلات دینامیکی، از نرم‌افزار سیم‌مکانیک نیز استفاده شده‌است (شکل ۲۳). مدل اصطکاک ویسکوز در این نرم‌افزار در شکل (۲٤) نشان داده شده‌است.



شکل ۲۳ مدل ربات مار مانند با ۸ لینک در نرم‌افزار سیم‌مکانیک



شکل ۲٤ مدل اصطکاک ویسکوز در نرم‌افزار سیم‌مکانیک